

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan penelitian mengenai program deteksi *genomic repeats* dengan implementasi algoritma Knuth-Morris-Pratt pada R *package high-performance computing pbdMPI*, maka penulis mendapatkan beberapa kesimpulan yang berhubungan dengan tujuan penelitian. Berikut kesimpulan yang dapat penulis jabarkan.

1. Berhasil membuat model *parallel computing* untuk pencarian *pattern* pada *string* menggunakan algoritma Knuth-Morris-Pratt. Model tersebut memiliki konsep pemotongan *string* agar tidak terjadi *miss* ketika *pattern* yang dicari tepat berada pada pemotongan *string*. Pada model tersebut juga dikenalkan variabel *adder* sebagai penambah nilai *index* dari setiap hasil keluaran komputasi secara paralel agar sesuai dengan *index* pada *string* yang utuh.
2. Berhasil mengimplementasi program dengan model yang telah dirancang sebelumnya untuk meningkatkan kecepatan dan menjaga akurasi hasil yang baik. Program yang dibangun menggunakan bahasa pemrograman R dengan *package pbdMPI* untuk melakukan *parallel computing*. Program ini telah diuji untuk *error handling* dan akurasinya terhadap data yang lebih kecil.
3. Telah melakukan total 117 eksperimen yang terbagi ke dalam skenario *stand alone* sebanyak 9 eksperimen dan skenario *parallel computing* sebanyak 108 eksperimen. Eksperimen tersebut dilakukan pada satu komputer dengan spesifikasi yang sama. Eksperimen dengan waktu terlama adalah pencarian ‘CAG’ pada file kromosom 1 secara *stand alone* yang menghabiskan waktu komputasi sebanyak 81.439,6 detik atau sekitar 22,6 jam.
4. Melakukan analisa hasil eksperimen dengan kesimpulan bahwa pemangkasan waktu pemrosesan tidak beriringan dengan jumlah *core* yang digunakan dalam hal rasio. Contoh dalam penggunaan 4 *cores* tidak berarti memiliki waktu pemrosesan 50% dari waktu yang dibutuhkan jika menggunakan 2 *cores*. Hal ini disebabkan dengan adanya proses pembacaan file, praproses

pembersihan *string* dan juga pencarian *pattern* yang berulang berdekatan. Selain itu nilai *iterator* yang optimal bergantung pada panjang karakter dari *string* mengingat akan terjadinya komunikasi yang banyak pula jika *iterator* diatur dengan nilai yang sangat besar. Selain sumber daya, panjang karakter *string*, *prefix pattern* dan nilai *iterator* ternyata jumlah *pattern* yang ditemukan pada *string* juga mempengaruhi waktu komputasi karena adanya proses penyimpanan *index* dan juga pencarian *pattern* berulang yang akan semakin lama jika *pattern* yang ditemukan pada *string* semakin banyak.

5.2 Saran

Dalam pelaksanaan penelitian, penulis menyadari bahwa masih banyak kekurangan yang dilakukan oleh penulis dalam penelitian ini. Oleh karena itu, penulis menyampaikan beberapa saran yang dapat dilakukan di kemudian hari. Penelitian selanjutnya dapat menghasilkan sebuah program yang jauh lebih baik dalam hal kecepatan. Berikut beberapa saran yang dapat penulis anjurkan.

1. Dalam input data seperti nama file, *pattern* dan nilai *iterator* dapat dimasukkan pengguna pada *terminal/prompt* karena kemungkinan akan adanya kesulitan pengguna dalam memasukkannya pada kode program.
2. Meneliti penggunaan sumber daya untuk eksekusi program agar dapat dilakukan pada sumber daya komputer seminimal mungkin.
3. Sebaiknya dapat membuat sebuah model untuk pembacaan file secara paralel juga mengingat tahap ini dilakukan secara *stand alone* pada penelitian ini.
4. Membuat beberapa fungsi praproses yang lebih beragam dan dinamis sehingga dapat digunakan pada data yang universal, tidak berpatok pada data dengan standar format NCBI/Ensembl.
5. Dapat membuat model atau formula untuk menentukan *nilai* *iterator* optimal.
6. Penulis berharap program ini dapat digunakan untuk menunjang perkembangan ilmu pengetahuan khususnya dalam bidang ilmu komputer dan biologi.